

94 JSON Interview Questions to Hire Top Developers

Questions

1. What is JSON? Can you explain it like I'm five?
2. Why do we use JSON instead of other formats?
3. What are the basic building blocks of JSON data (like, the stuff you see inside)?
4. Can you give me an example of a simple JSON object?
5. How do you show a list of things in JSON?
6. What does it mean if a JSON value is 'null'?
7. How are JSON objects different from JSON arrays?
8. What's wrong with this JSON: {name: John, age: 30}?
9. If you have a JSON string, how can you see what's inside it using code?
10. Can JSON store complicated things like pictures or videos directly? If not, how can it handle such data?
11. Imagine you want to send your friend's name, age, and favorite colors as JSON. How would you do that?
12. Is JSON just for websites, or can other programs use it too?
13. What happens if you accidentally forget a comma in your JSON? (Imagine the drama!)
14. How do different programming languages (like Javascript and Python) work with JSON?
15. If you get a JSON error, what are some things you can check to fix it?
16. Let's say you receive a JSON with information you don't need. Is it okay to just ignore it, or do you have to handle it somehow?
17. Explain the use of JSON schema. Why is it important?
18. How do you represent nested objects in JSON? Give an example.
19. Describe a real-world scenario where JSON is commonly used.
20. What are the data types supported by JSON?
21. How does JSON handle Unicode characters? Why is this important?
22. If you need to send a date in JSON, what's a common way to format it?
23. Can you have multiple root elements in a JSON document? Why or why not?
24. What's the difference between JSON and XML? When would you use one over the other?
25. How can you validate if a given string is a valid JSON?
26. Explain JSONP and why it was used, and if there are better options now.
27. What is a JSON parser and what does it do?
28. What is JSON? Can you describe it like you're explaining it to a five-year-old?
29. Have you ever seen JSON before? Where?
30. What kinds of things can JSON hold inside it? Like, can it hold numbers, words, or other things?
31. What are the curly braces {} and square brackets [] used for in JSON?
32. Can you spot the error in this JSON: {'name': 'John', 'age': 30}? Why is it wrong?
33. What does 'key' and 'value' mean in JSON, using a simple example?
34. If you have a list of friends, how could you represent it using JSON?
35. What is the difference between a JSON object and a JSON array?
36. Why do we use double quotes " around keys in JSON?
37. Imagine you want to send information about a toy car over the internet. How could you describe it using JSON?
38. Is this valid JSON: [1, 2, 3]? Why or why not?
39. Can JSON hold different types of data in the same object, like a number and a word?
40. What is null in JSON? What does it mean?
41. How would you represent a list of books with their titles and authors in JSON?
42. If someone says they are 'parsing' JSON, what are they doing?
43. Why is JSON useful for sending information between computers?
44. Can you give an example of a simple JSON object that contains your name and your favorite color?
45. What happens if you have two keys with the same name in a JSON object? Is that allowed?
46. Is a JSON document just one big line of text, or can it be formatted to be easier to read?
47. Can you describe a situation where you might use a JSON array inside a JSON object?
48. Explain the difference between JSON and JSONP, and why JSONP was developed.
49. How can you handle errors when parsing JSON data in JavaScript?
50. Describe a scenario where you might need to serialize a JavaScript object into JSON.
51. What are the advantages of using JSON over XML for data transmission?
52. How do you validate a JSON document against a predefined schema?
53. Explain how to use JSON with AJAX to retrieve data from a server.
54. What security considerations should you keep in mind when working with JSON data from untrusted sources?
55. How can you minimize the size of a JSON payload to improve performance?
56. Explain the concept of JSON Schema and its purpose.
57. How do you handle circular references when serializing an object to JSON?
58. What are some common JSON parsing libraries available in Python, and how do they differ?
59. How can you transform a JSON document into a different format using a tool like jq?
60. Describe a situation where you would use JSON Web Tokens (JWT) for authentication.
61. How do you handle nested JSON objects when extracting specific data?
62. Explain how to use JSON with Node.js to create a simple API endpoint.
63. What is the purpose of the Content-Type header when sending JSON data in an HTTP request?
64. How can you pretty-print a JSON string in Python for better readability?
65. Describe a scenario where you would need to flatten a nested JSON structure.
66. How do you handle missing or null values in a JSON document when processing it?
67. Explain how to use JSON with PHP to decode data received from a client.
68. What are the limitations of using JSON.stringify() for complex objects?
69. How can you ensure that a JSON API is RESTful in its design?
70. Describe how you would debug issues related to JSON parsing or serialization.
71. How can you use JSON to represent configuration data for an application?
72. Explain the difference between JSON array and JSON object.
73. What are some tools to validate JSON?
74. How does JSON relate to NoSQL databases like MongoDB?
75. Describe a complex JSON schema you've designed and how you ensured its validity and maintainability over time.
76. How would you handle a situation where a JSON API you depend on introduces breaking changes? Discuss strategies for versioning and graceful degradation.
77. Explain your approach to optimizing JSON data transfer for mobile applications, considering factors like payload size and parsing performance.
78. Discuss the trade-offs between using JSON and other data serialization formats like Protocol Buffers or Avro in a high-performance system.
79. How do you approach testing JSON APIs, including validation of request and response formats, error handling, and security considerations?
80. Describe a time you used JSON Schema to validate data, and how it improved the development process.
81. Explain how you would design a system to handle large JSON datasets efficiently, considering storage, querying, and indexing.
82. How do you manage different versions or schemas of JSON documents in a system that requires backward compatibility?
83. Discuss your experience with JSON transformation tools (e.g., jq) and how they can be used to manipulate JSON data effectively.
84. How would you secure a JSON API against common attacks like JSON injection or cross-site scripting (XSS)?
85. Describe strategies for handling nested JSON structures and how to efficiently access and modify deeply nested values.
86. Explain how you would implement a JSON-based configuration system for a distributed application, including considerations for dynamic updates and consistency.
87. Discuss your experience with using JSON Web Tokens (JWT) for authentication and authorization in a web application.
88. How do you approach debugging issues related to JSON serialization or deserialization in a complex application?
89. Describe a project where you used JSON to represent complex data relationships, and the challenges you faced in modeling those relationships.
90. Explain how you would monitor the performance of a JSON API, including metrics like response time, error rates, and resource utilization.
91. How do you handle binary data within a JSON structure? What encoding methods are appropriate and why?
92. Discuss how JSON Patch or JSON Merge Patch can be used to update parts of a JSON document efficiently, and the advantages of using these techniques.
93. Explain your understanding of JSON-LD and how it can be used to represent linked data on the web.